1. (10 points) The uniform distribution for a continuous variable x is

$$p(x; a, b) = \frac{1}{b-a}, \qquad a \le x \le b.$$

Verify that this distribution is normalized (integrats to one), and find expressions for its mean and variance.

Ans-1 = Any distribution function is normalized if it integrates to 1 over its domain.

$$p(x; a, b) = \frac{1}{b-a} \quad \text{where} \quad a \le x \le b$$

$$\int_a^b p(x; a, b) = \int_a^b \frac{1}{b-a}\, dx$$

$$= \frac{1}{b-a} \left[x\right]_a^b$$

$$= \frac{b-a}{b-a}$$

$$= 1$$

Hence this distribution is normalized.

Mean =

$$E(x) = \int_a^b x \cdot p(x)\, dx$$

$$= \int_a^b x \cdot \frac{1}{b-a}\, dx$$

$$= \frac{1}{b-a} \left[\frac{x^2}{2}\right]_a^b$$

$$= \frac{b^2-a^2}{2(b-a)}$$

$$= \frac{(b-a)(b+a)}{2(b-a)}$$

$$= \frac{a+b}{2}$$

Variance =

$$E\left[(X - E[X^2])\right] = E[X^2] - (E(X))^2$$

$$\Rightarrow \int_a^b \frac{1}{b-a} x^2 \cdot dx - \left(\frac{a+b}{2}\right)^2$$

[E[X] = mean we found earlier]

$$\Rightarrow \frac{1}{b-a}\left[\frac{x^3}{3}\right]_a^b - \left(\frac{a+b}{2}\right)^2$$

$$= \frac{b^3 - a^3}{3(b-a)} - \left(\frac{a+b}{2}\right)^2$$

$$= \frac{(b-a)(b^2+ab+a^2)}{3(b-a)} - \left(\frac{a^2+b^2+2ab}{4}\right)$$

$$= \frac{b^2+ab+a^2}{3} - \left(\frac{a^2+b^2+2ab}{4}\right)$$

$$= \frac{4b^2+4ab+4a^2-3a^2-3b^2-6ab}{12}$$

$$= \frac{a^2+b^2-2ab}{12} = \frac{(a-b)^2}{12}$$

2. (10 points) Recall that the PMF of a Poisson random variable is

$$p(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!},$$

with one parameter λ. Given i.i.d. samples $x_1, ..., x_n \sim \text{Pois}(\lambda)$, derive the MLE for λ.

Ans:2 =

PMF of poinon random variable is =

$$P(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

Taking log likelihood =

$$l(\lambda) = \ln \sum_{i=1}^{n} \frac{\lambda^x e^{-\lambda}}{x_i!}$$

$$= \sum_{i=1}^{n} \ln \frac{e^{-\lambda}\lambda^x}{x_i!}$$

$$= \sum_{i=1}^{n} (x_i \log\lambda - \lambda - \log x_i!)$$

$$= \log\lambda \sum_{i=1}^{n} x_i - n\lambda - \sum_{i=1}^{n} \log x_i!$$

Equating derivative to 0 $\Rightarrow$

$$l'(\lambda) = \frac{d}{d\lambda} l(\lambda) = 0$$

$$\Rightarrow \frac{d}{d\lambda} \left\{ \log \lambda \sum_{i=1}^{n} x_i - n\lambda - \sum_{i=1}^{n} \log x_i ! \right\} = 0$$

$$\Rightarrow \frac{1}{\lambda} \sum_{i=1}^{n} x_i - n = 0$$

Therefore MLE = $\frac{\sum_{i=1}^{n} x_i}{n}$

3. (10 points) The function `randn(d,1)` generates a multivariate normal variable $\boldsymbol{x} \in \mathbb{R}^d$ with zero mean and covariance $\boldsymbol{I}$. Describe how to generate a random variable from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. *Hint.* If $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\boldsymbol{Ax} + \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{A\mu} + \boldsymbol{b}, \boldsymbol{A\Sigma A}^\top)$.

Ans-3 = An $x \sim N(0,1)$

now as we have a matrix A and a normal variable x. we will generate a random variable y such that $y = Ax + u$ and verify that $y \sim N(u, \Sigma)$

$x \sim N(u, \Sigma)$,

then $Ax+b \sim N(Au+b, A\Sigma A^T)$

Therefore for $x \sim N(0, I)$

Then $y = Ax + u \sim N(A*0+u, AIA^T)$

Hence $y \sim N(u, AA^T) = N(u, \Sigma)$

4. (20 points) Consider a data set in which each data sample i is associated with a weighting factor $r_i > 0$, and we instead try to minimize the weighted MSE function

$$\underset{\boldsymbol{\theta}}{\text{minimize}} \quad \frac{1}{n}\sum_{i=1}^{n} r_i(y_i - \boldsymbol{\phi}_i^\top \boldsymbol{\theta})^2.$$

Find an expression for the solution $\widehat{\boldsymbol{\theta}}$ that minimizes this loss function.

Ans-4 = $\underset{\theta}{\text{minimize}} \; \frac{1}{n} \sum_{i=1}^{n} r_i \left(y_i - \phi_i^T \theta\right)^2$

To minimize this weighted MSE function we have; $\hat{\theta} = \left(\phi^T \phi\right)^{-1} \phi^T \psi$

where,

$$\psi = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \text{and} \quad \phi = \begin{bmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_n^T \end{bmatrix}$$

Therefore we can write ϕ and ψ as =

$$\psi' = \begin{bmatrix} \sqrt{r_1}\, y_1 \\ \sqrt{r_2}\, y_2 \\ \vdots \\ \sqrt{r_n}\, y_n \end{bmatrix} = P^T \psi \quad \text{and} \quad \phi' = \begin{bmatrix} \sqrt{r_1}\, \phi_1^T \\ \sqrt{r_2}\, \phi_2^T \\ \vdots \\ \sqrt{r_n}\, \phi_n^T \end{bmatrix} = P^T \phi$$

where $P = \begin{bmatrix} \sqrt{\sigma_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_n} \end{bmatrix}$

Hence $\hat{\omega} = (\phi'^T \phi')^{-1} \phi'^T \psi'$

$= \left((P^T \phi)^T (P^T \phi) \right)^{-1} (P^T \phi)^T (P^T \psi)$

$= \left(\phi^T P . P^T \phi \right)^{-1} \left(\phi^T P . P^T \psi \right)$

where,

$P . P^T = \begin{bmatrix} \sqrt{\sigma_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_n} \end{bmatrix} \begin{bmatrix} \sqrt{\sigma_1} & 0 & \cdots & 0 \\ 0 & \sqrt{\sigma_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sqrt{\sigma_n} \end{bmatrix}$

$P . P^T = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix}$

5. (50 points) The 20 Newsgroups data set is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. The data set can be downloaded here: <http://qwone.com/~jason/20Newsgroups/>. For simplicity, we will just focus on the "bag-of-words" representation of the documents given in the Matlab/Octave section. In this case, the input $\boldsymbol{x}_i$ is a vector of word histogram of doc i, and the output y_i is the news group that it belongs to.

   (a) One effective classifer by Tom Mitchell is an instance of the naive Bayes model. First, the actual word count is ignored in the input data; we only consider whether a word j appears in doc i or not. Then each feature in $\boldsymbol{x}$ can be viewed as a Bernoulli random variable. Furthermore, the naive Bayes assumption states that each of these Bernoulli random variables are conditionally independent given the label y, i.e., $p(\boldsymbol{x}|y) = \prod p(x_j|y)$. Each of the $p(x_j|y)$ can be easily estimated using the training data.

Ans. (a) =

Derivation for mathematical expression for Bayes model.

$x \in \{0, 1\}^K$

$y \sim \text{Bernoulli}(\phi)$

$x_k \sim \text{Bernoulli}(\theta_{k|y}) \quad \forall k \in \{1, \dots K\}$

$\text{Bern}(x|\theta) = P(x|\theta) = \theta^x (1-\theta)^{1-x}$

$D = \{x_1, x_2 \dots \dots x_N\}$

$P(\theta|x) \propto P(\theta) P(x|\theta)$

$$P(\theta|a,b) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(a,b)}$$

$$P(\theta|D) \propto \theta^{\alpha-1}(1-\theta)^{\beta-1}\theta^{n_1}(1-\theta)^{n_2}$$

$$P(\theta|D) \propto \theta^{n_1+a-1}(1-\theta)^{n_2+\beta-1}$$

$$P(\theta|D) = \frac{\theta^{n_1+a-1}(1-\theta)^{n_2+\beta-1}}{B(a+n_1+b+n_2)}$$

$$\hat{\theta}_{map} = \frac{\alpha+n_1-1}{\alpha+\beta+n_1+n_2-2} = \frac{n_1+1}{N+2}$$

N = number of documents in class that contain feature n_1

n is the count of feature in class after getting $\hat{\theta}_{map}$ for all the features in vocabulary. we extract features for test data. we multiply the features with all the corresponding $\hat{\theta}_{map}$

for all values of k where k is the number of class.

The class score with maximum value is assigned as class of the data.

$$\hat{\theta}_{map} = \arg\max_{\theta} \prod_{i=1}^{N} P(X^{(i)} | \theta) P(\theta)$$

(b) To incorporate the word count, we can impose a rather different probabilistic model. Assume each doc is a huge multinomial random variable, with cardinality equal to the vocabulary size and the total number of draws is the length of that doc, given the label. In other words, $p(\boldsymbol{x}|y)$ is multinomial.

Ans.b = we know that,

$$P(c|d) = \frac{P(d|c)\,P(c)}{P(d)}$$

MLE of C =>

$$C_{Map} = \operatorname{argmax}_{c \in C} P(c|d)$$

$$= \operatorname{argmax}_{c \in C} \frac{P(d|c)\,P(c)}{P(d)}$$

$$= \operatorname{argmax}_{c \in C} P(d|c)\,P(c)$$

$$= \operatorname{argmax}_{c \in C} P(x_1, x_2 \dots x_n|c)\,P(c)$$

$$C_{NB} = \operatorname{argmax}_{c \in C} P(c_j) \prod_{x \in X} P(x|c)$$

$$= \operatorname{argmax}_{c \in C} P(c_j) \prod_{i \in \text{position}} P(x_i|c_j)$$

$$\hat{P}(c_j) = \frac{\text{count documents}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

Parameter Smoothing =

$$\hat{P}(w_i | c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

Laplace Smoothing =

$$\hat{P}(w_i | c) = \frac{\text{count}(w_i, c_j) + 1}{\sum_{w \in V} (\text{count}(w, c_j) + 1)}$$

$$= \frac{\text{count}(w_i, c_j) + 1}{\sum_{w \in V} (\text{count}(w, c_j) + |V|)}$$